CLAIMS

What is claimed is:

1. A method for statically detecting a datarace in a multithreaded application, said method comprising:

inputting a set of input information;

processing the set of input information; and

outputting a statement conflict set that identifies the statement pairs whose execution instances definitely or potentially cause dataraces, without executing the multithreaded application.

2. The method of claim 1, wherein the processing comprises:

selectively evaluating the input information with an IsPotentialDR relation; and

selectively evaluating the input information with an IsDefiniteDR relation.

3. The method of claim 2, wherein, for a given pair of reference expressions, the IsPotentialDR relation comprises:

determining whether the reference expressions might be executed by different threads (negation of DefSameThreadObj);

determining whether the reference expressions might access the same field of the same object; and

determining whether the reference expressions might not be mutually synchronized (negation of DefSync).

4.   The method of claim 2, wherein, for a given pair of reference expression, the IsDefiniteDR relation comprises:

determining whether the reference expressions cannot be executed by the same thread (negation of PossSameThreadObj);

determining whether the reference expressions must access the same field of the same object;

determining whether the reference expressions cannot be mutually synchronized (negation of PossSync); and

determining whether the reference expressions must execute.

5.   The method of claim 1, wherein the set of input information comprises a multithreaded context graph (multithreaded context graphs).

6.   The method of claim 1, wherein the multithreaded context graphs comprises an interprocedural call graph having each of a plurality of synchronized blocks as a separate node.

7.   The method of claim 1, wherein the multithreaded context graph

comprises an interprocedural call graph having each of a plurality of synchronized

methods as a separate node.

8.   The method of claim 1, further comprising

performing dynamic datarace detection on the statement conflict set.

9.   The method of claim 1, further comprising performing escape analysis to

identify statements that can access memory locations accessible by more than one

thread.

10.   The method of claim 1, wherein the processing comprises:

computing a node conflict set; and

computing the statement conflict set by determining pairs of conflicting

statements in the node conflict set.

11.   The method of claim 10, wherein the node conflict set computing

comprises:

initializing a synchronization object set for each of a plurality of

multithreaded context graph node.

12.   The method of claim 11, wherein the node conflict set computing further comprises:

        identifying all reachable conflicting node pairs for each thread-root node.

13.   The method of claim 12, wherein the node conflict set computing further comprises:

        identifying all reachable conflicting node pairs for each distinct pair of thread-root nodes in the multithreaded context graph; and

        identifying all reachable conflicting node pairs for each thread-root node in the multithreaded context graphs that is invokeable by more than one thread.

14.   The method of claim 1, wherein the input comprises meta-information relating to a multithreaded application written in an object-oriented programming language.

15.   The method of claim 1, wherein the input comprises an multithreaded context graph for a multithreaded application written in an object oriented programming language.

16.   The method of claim 15, wherein the input further comprises a plurality of bytecodes that collectively comprise the application.

17.   A computer processing system for statically detecting a datarace in a multithreaded application, comprising:

an input interface;

an output interface;

a storage medium comprising the application and meta-information relating to the application; and

a processor configured to receive the application and the meta-information, process the application and the meta-information without executing the application, and determine a statement conflict set (SCS) for the application.

18.   The computer processing system of claim 17, wherein the meta-information comprises a multithreaded context graph.

19.   The computer processing system of claim 17, wherein the processor is further configured to perform dynamic datarace detection on the statement conflict set.

20.   A computer program product, comprising a computer readable medium having computer code embodied therein for statically detecting a datarace in a

multithreaded application, said computer program product comprising:

computer readable program code devices configured to receive the application and the meta-information;

computer readable program code devices configured to process the application and the meta-information without executing the application; and

computer readable program code devices configured to determine a statement conflict set (SCS) for the application.